



Reconfigurable, Intelligently-Adaptive, Communication System, an SDR Platform



SPACE COMMUNICATIONS AND NAVIGATION

Presented By:

Rigoberto Roche'
NASA Glenn Research Center

www.nasa.gov





Overview

- Introduction:

This section provides a streamlined overview of the hardware and functional characteristics of the space SDR

- Background Overview of the Work
- Hardware Components Off The Shelf (COTS)
- Software Defined Radio Functional Diagram

- SDR Implementation

Overview of the Software and Hardware Development Efforts and Testing Performed on the Space SDR

- General Purpose Module Software Architecture
- Command and Data Paths
- Command Hierarchy
- FPGA Transmit Side and Receive Side Wrapper
- FPGA Sample Waveform
- Radio Frequency Module
- Test Configuration for Data and Command Path Full Test

- Results and Conclusions

Overview of Accomplished Work, Relevance to NASA's mission and Overall Lessons Learned.



INTRODUCTION

Streamlined Overview of the Hardware and Functional Characteristics of the Space SDR



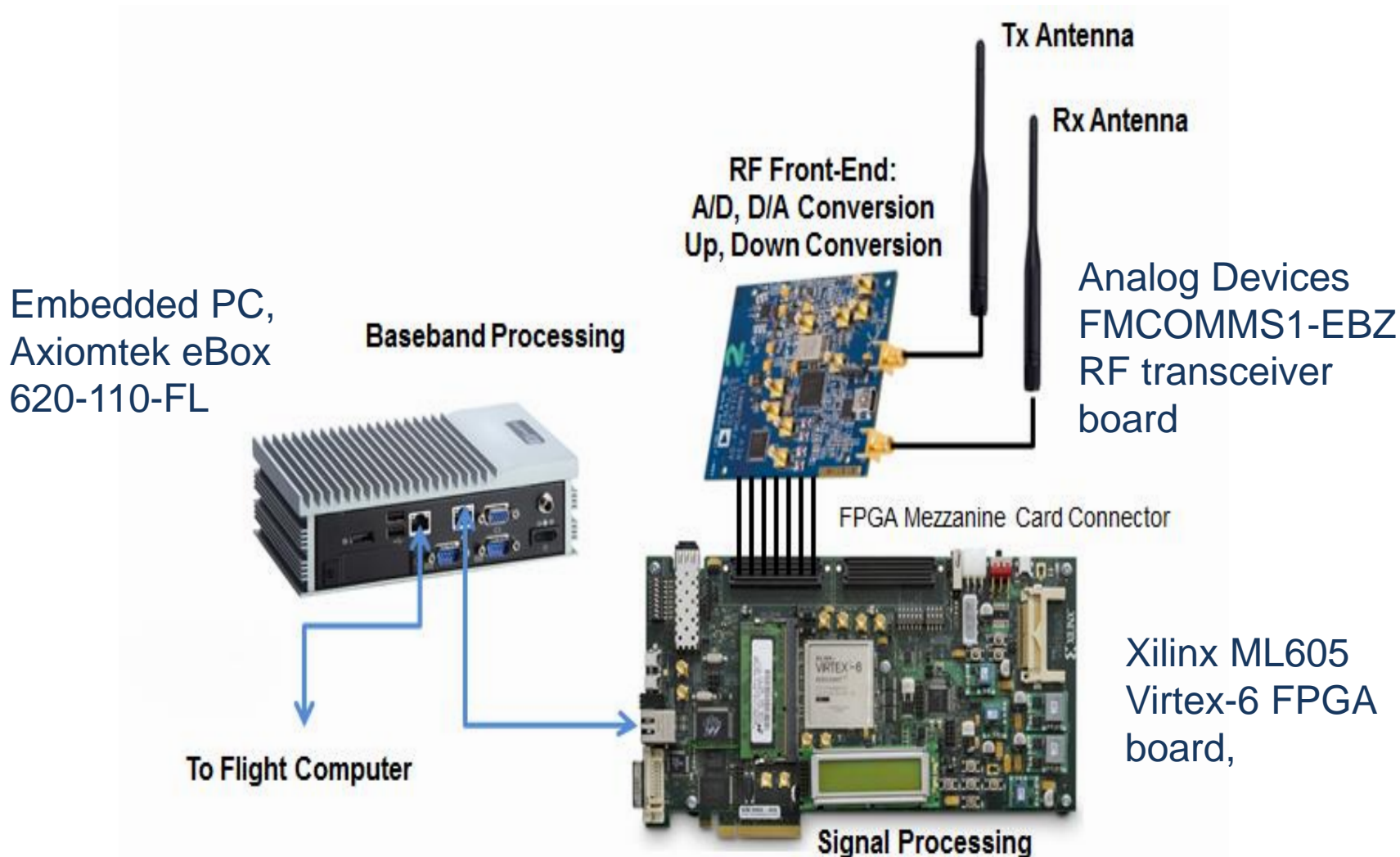
Introduction

Background Overview of the Work

- The Space Telecommunications Radio System (STRS) provides a common, consistent framework to abstract the application software from the radio platform hardware.
- STRS aims to reduce the cost and risk of using complex, configurable and reprogrammable radio systems across NASA missions.
- The NASA Glenn Research Center (GRC) team made a software defined radio (SDR) platform STRS compliant by adding an STRS operating environment and a Field programmable gate array (FPGA) wrapper, capable of implementing each of the platforms interfaces, as well as a test waveform to exercise those interfaces.
- This effort serves to provide a framework toward waveform development onto an STRS compliant platform to support future space communication systems for advanced exploration missions.

Introduction

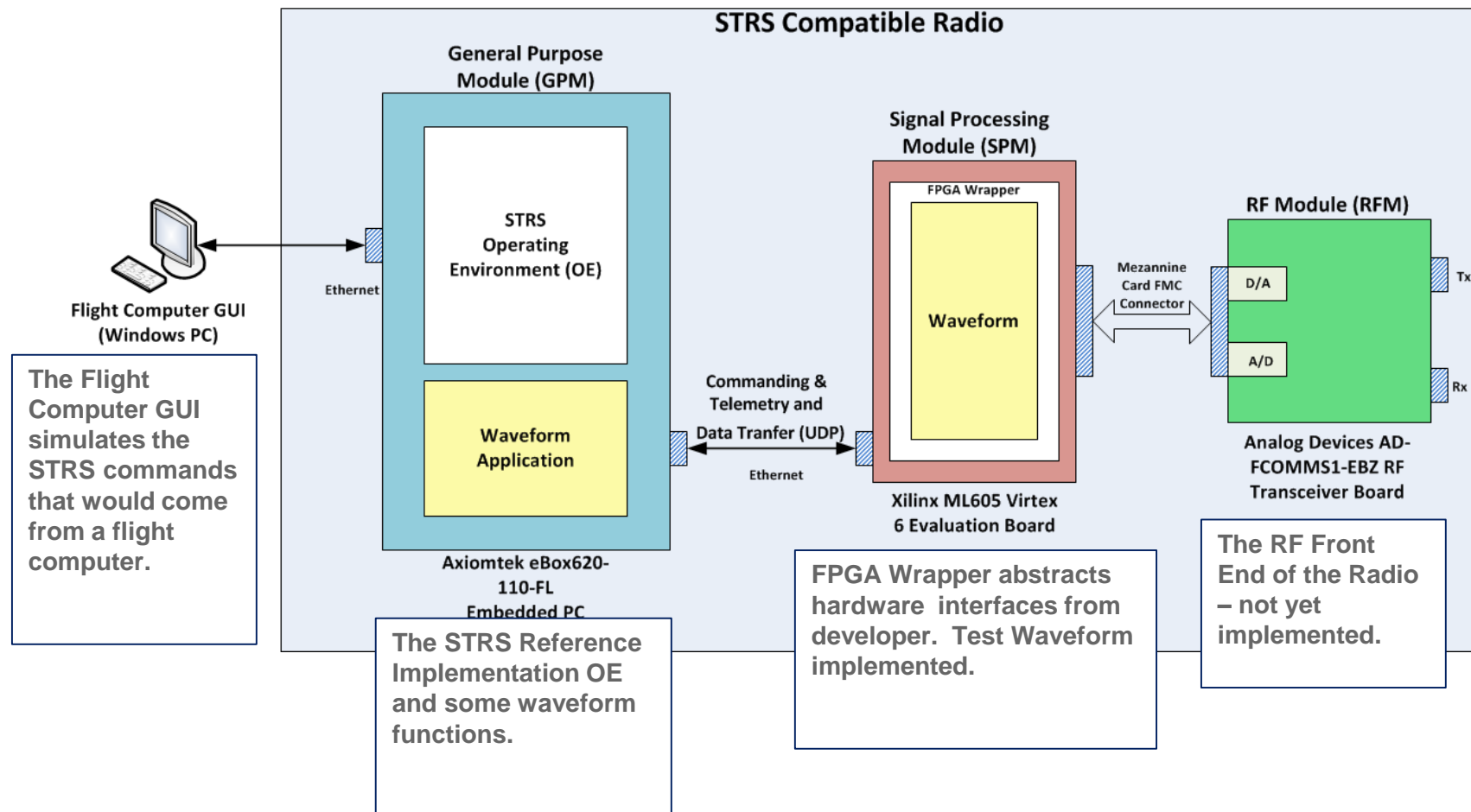
Hardware Components Off The Shelf (COTS)



Introduction

IPAS STRS Software Defined Radio Functional Diagram

- Diagram describes the functional components of the Implementation of the STRS Standard for the iPAS Space Radio.





SDR IMPLEMENTATION

**Overview of the Software and Hardware Development Efforts as
well as the Testing Performed on the Space SDR**



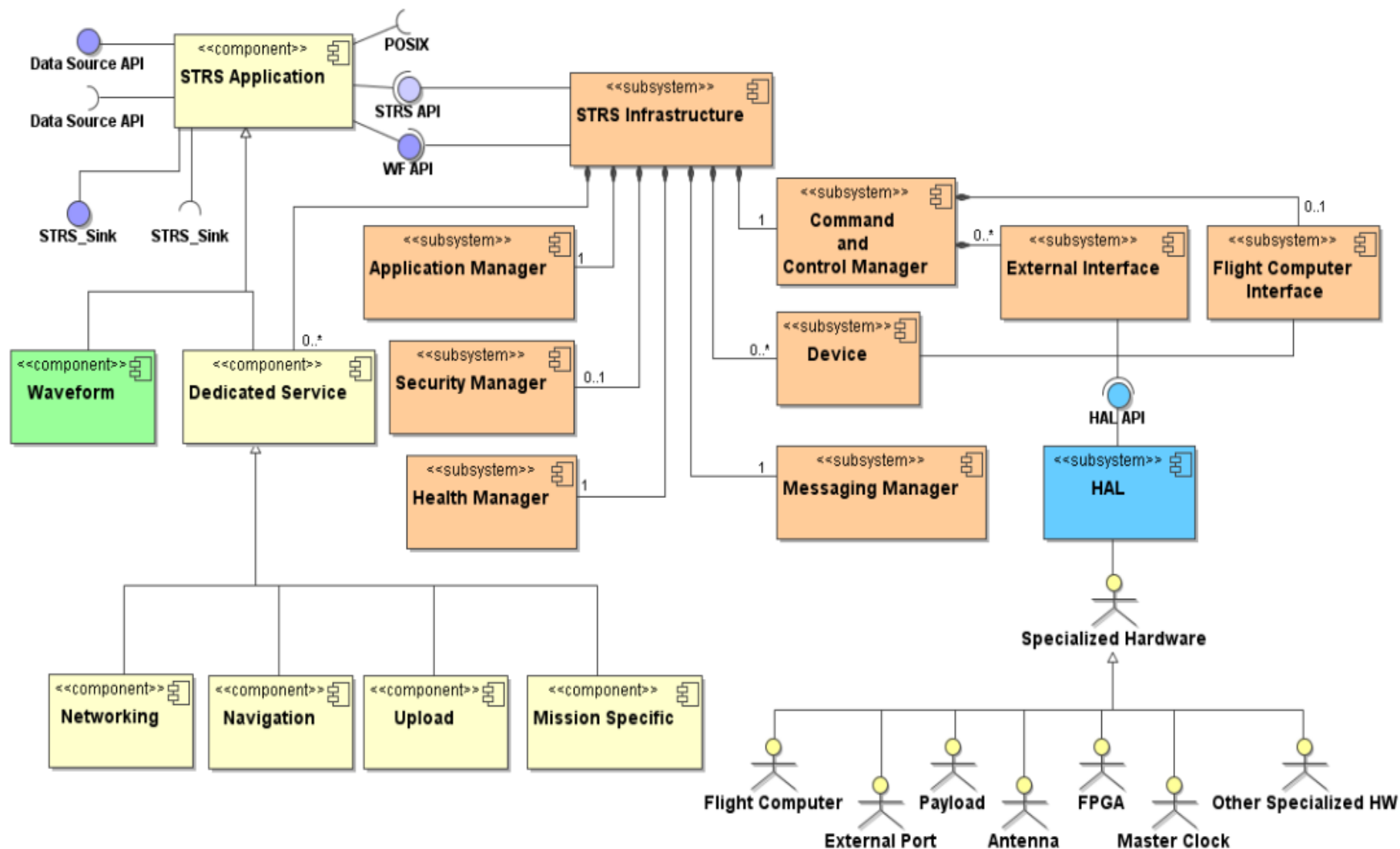
SDR Implementation

Software

- The STRS Reference Implementation (RI) is an application with a series of coded APIs, designed for plug-and-play.
- The RI serves as the framework for the instantiation and removal of waveforms and service components of the software, along with the execution of methods for each component, using well-defined interfaces to promote portability.
- This RI provides flexibility to use implementation standards for various mission classes, which can be used to create unique objects without having to recreate base classes.
- The RI also brings a set of test waveforms. These can be used to verify the expected behavior of a waveform within its current infrastructure.
- The STRS command infrastructure on the iPAS radio is implemented using the RI as a base framework, modified to work with the hardware of the SDR and to implement a sample waveform to control and test the SDR's interfaces.

SDR Implementation

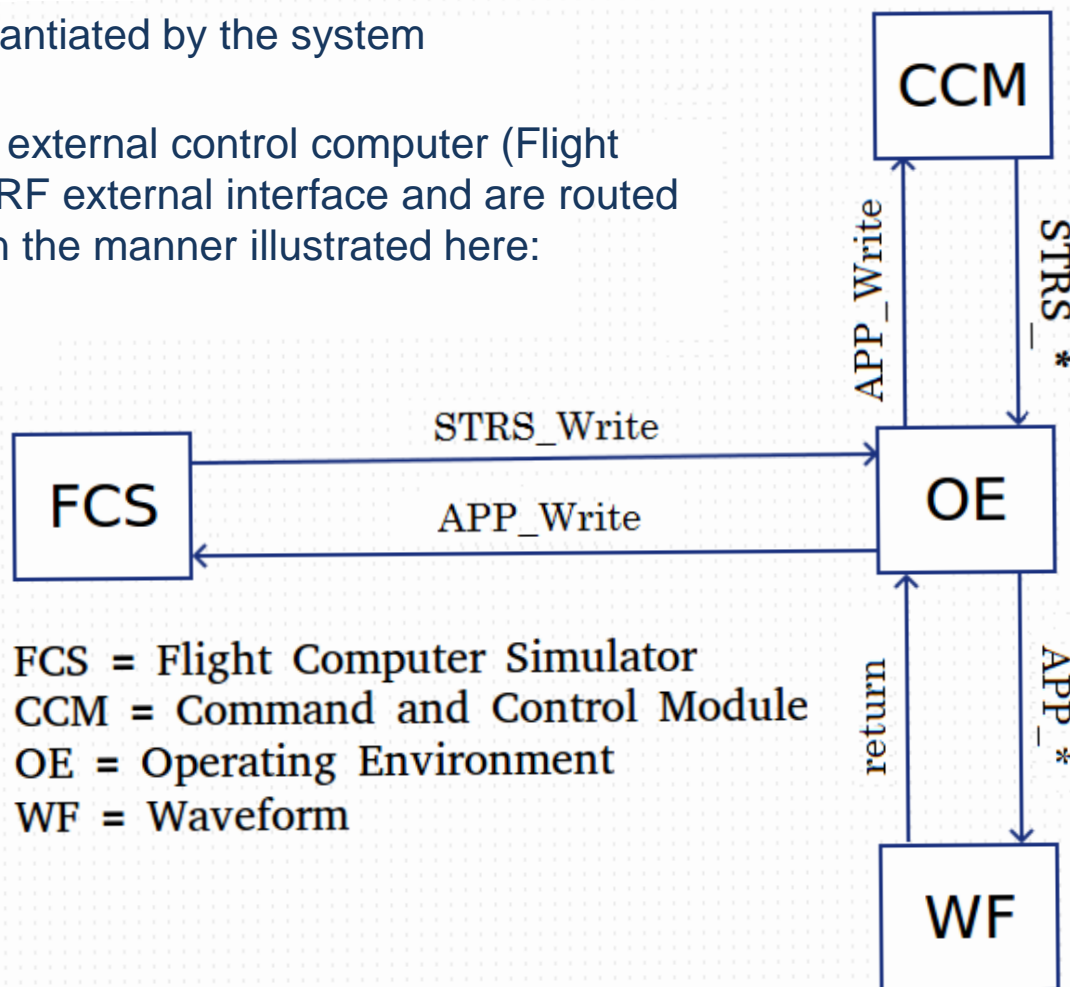
Software: General Purpose Module Software Architecture

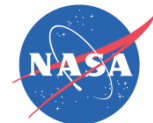


SDR Implementation

Software: STRS Architecture Command and Data Paths

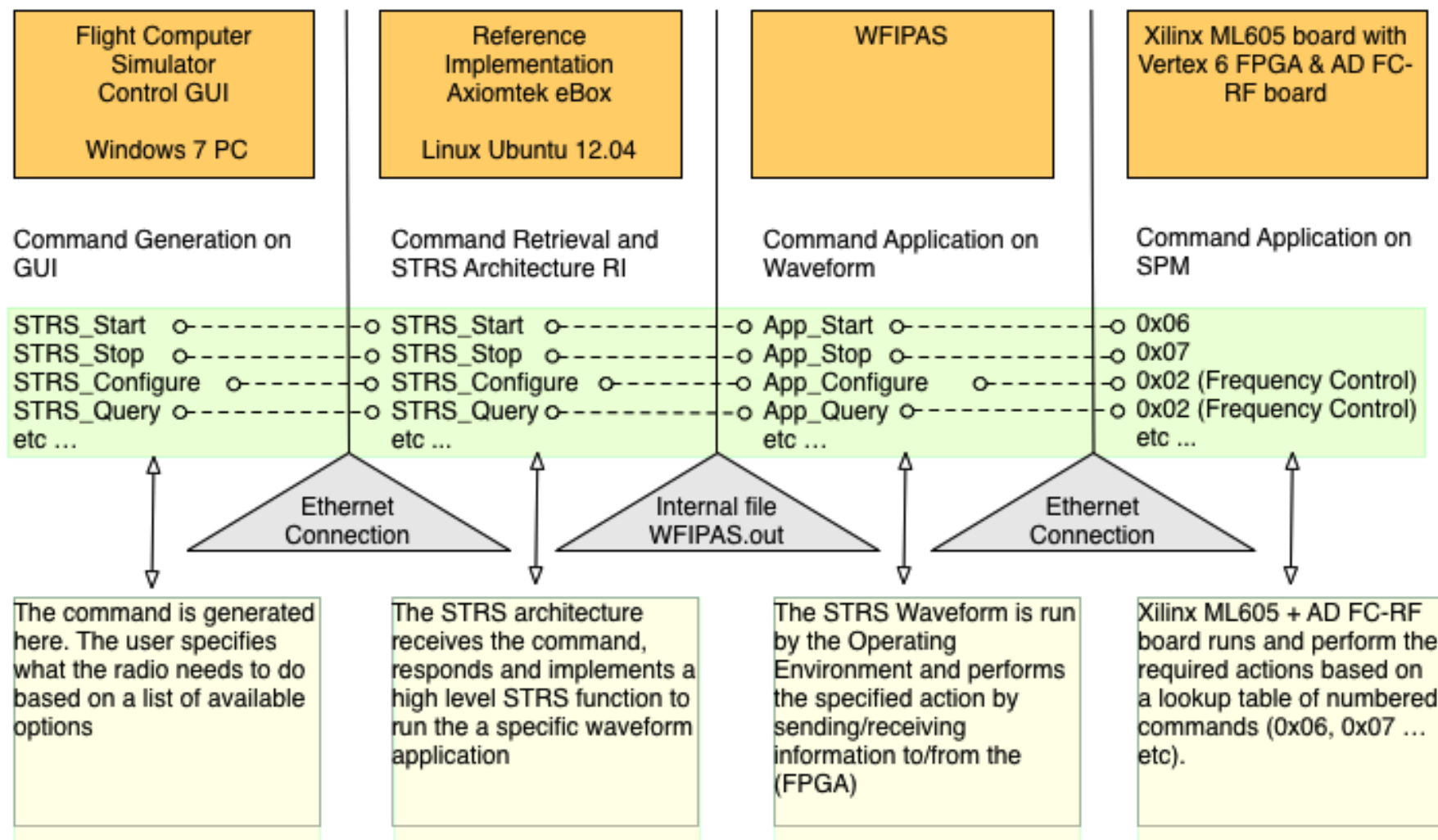
- The * represents a specific STRS or APP call to the application being instantiated by the system
- Call are made by the external control computer (Flight Computer) or by the RF external interface and are routed through the system in the manner illustrated here:





SDR Implementation

Software: STRS Architecture Command Hierarchy





SDR Implementation

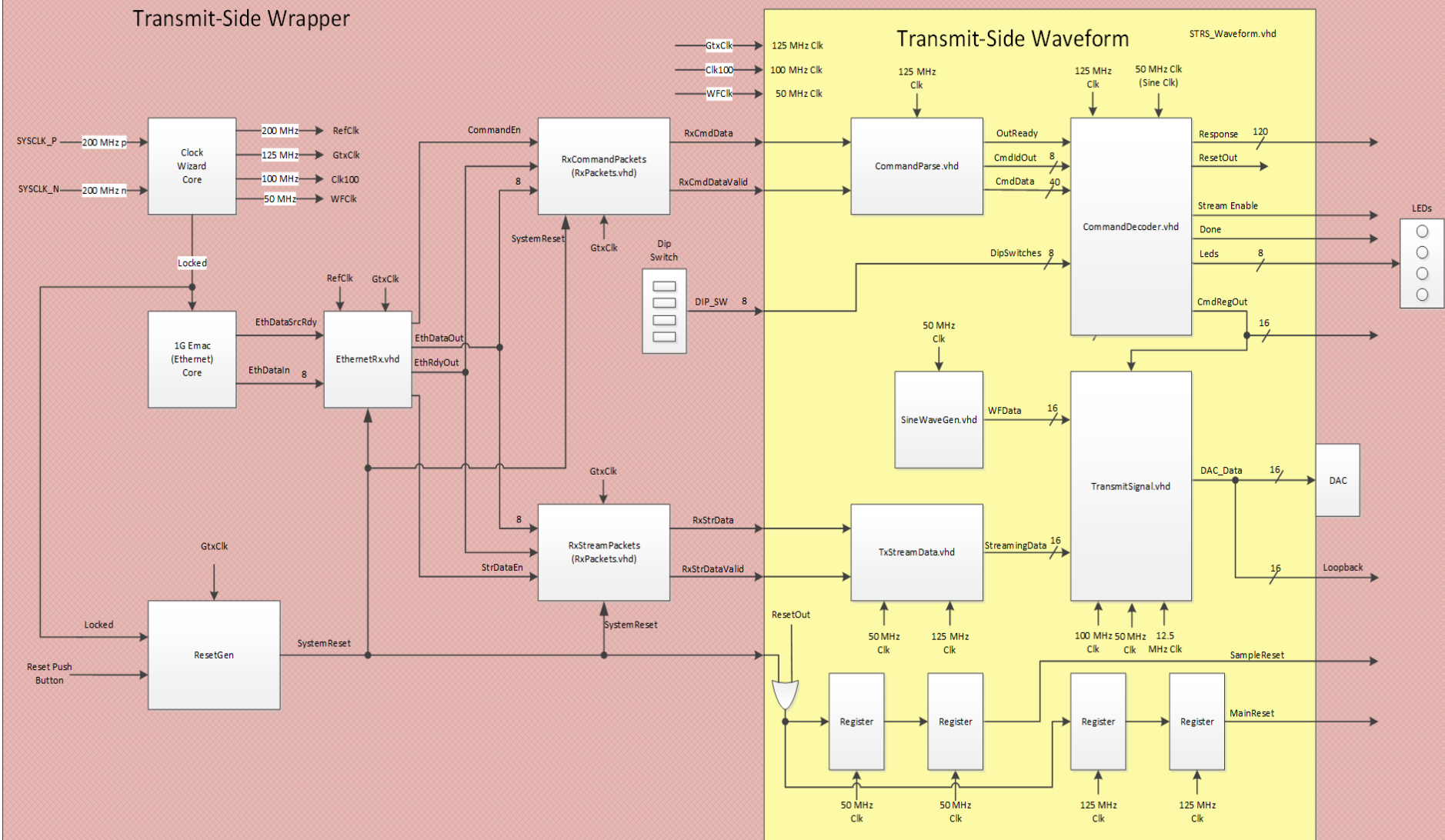
Firmware

- The firmware of the iPAS-STRS SDR is implemented on a Xilinx ML605, Rev D Evaluation board which contains a Virtex-6 FPGA. The development tool used for this design is Xilinx ISE Design Suite System Edition version 14.4
- STRS requires that the FPGA Wrapper for an STRS radio encompass all the possible radio FPGA interfaces abstracting these interfaces from the waveform, so that the waveform developer does not need to implement these interfaces manually.
- This approach also allows the platform developer to protect proprietary platform design information from a waveform developer, while including any other functionality that a radio would require, like power-on-resets and clock generation, which would be common to all radios on the platform.
- The wrapper also implements interfaces to on-board resources like clocks, switches, and LEDs, as well as the implementation of the interface to the Ethernet physical interface.

SDR Implementation

Firmware: FPGA Transmit Side Wrapper

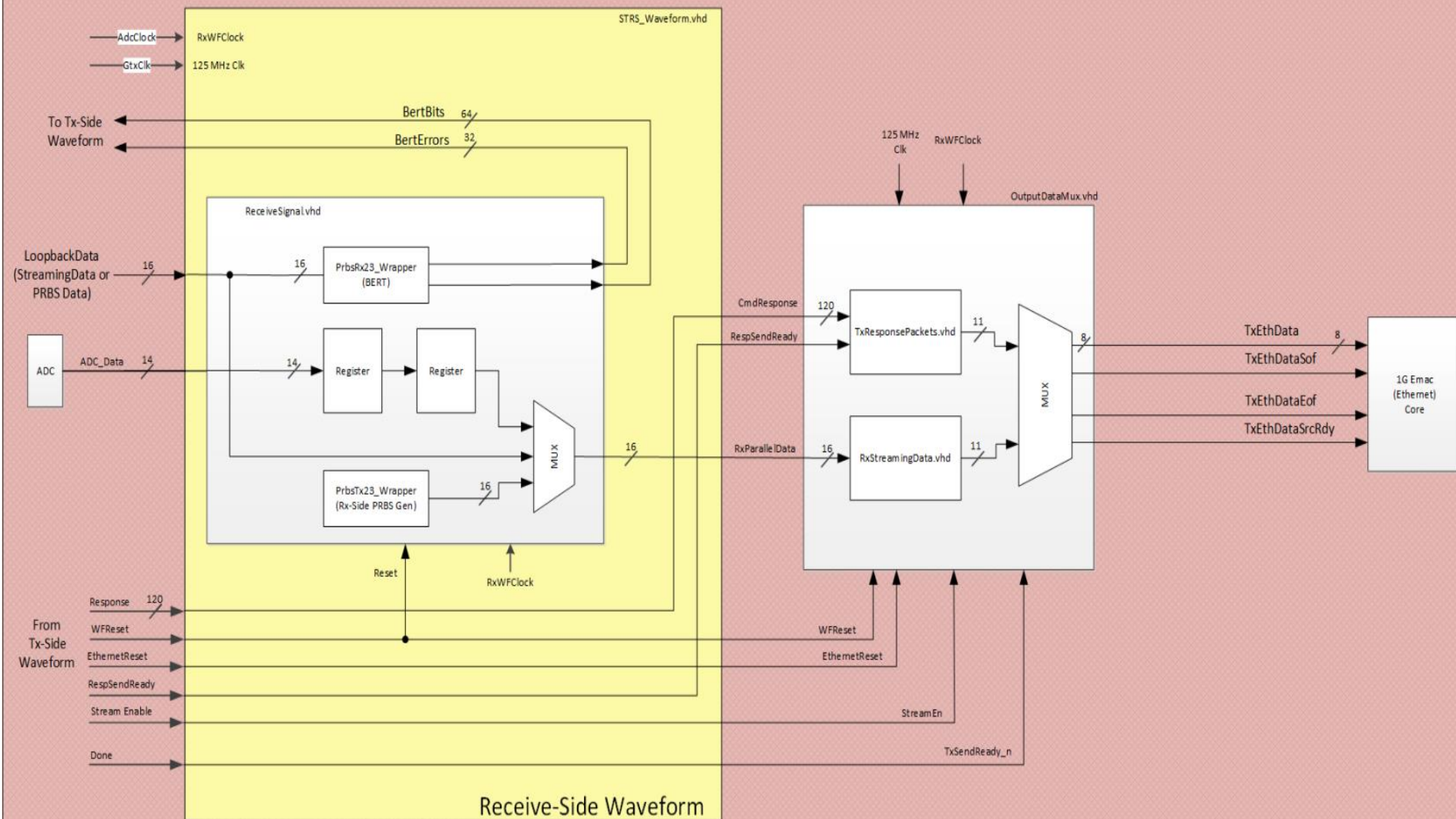
Transmit-Side Wrapper



SDR Implementation

Firmware: FPGA Receive Side Wrapper

Receive-Side Wrapper





SDR Implementation

Firmware: Radio Frequency Module

- The DAC and ADC for the iPAS STRS Radio are contained on the RF Module, Analog Devices FMCOMMS1-EBZ high-speed analog board.
- This is a mezzanine board, compatible with the Xilinx ML605 FPGA board, connected via the FMC-LPC connector.
- This RF Front-End board comes with a reference design implemented on the MicroBlaze soft processor, using Xilinx EDK and SDK.
- This reference design contains functionality not necessary for the STRS Radio implementation, so It was decided to use the reference design only for the configuration of the RF board, and not for the data paths.
- The FPGA wrapper, therefore, connects directly to the digital-to-analog and analog-to-digital converters directly with VHDL code. This approach greatly simplifies the FPGA wrapper and the insertion of new waveforms.



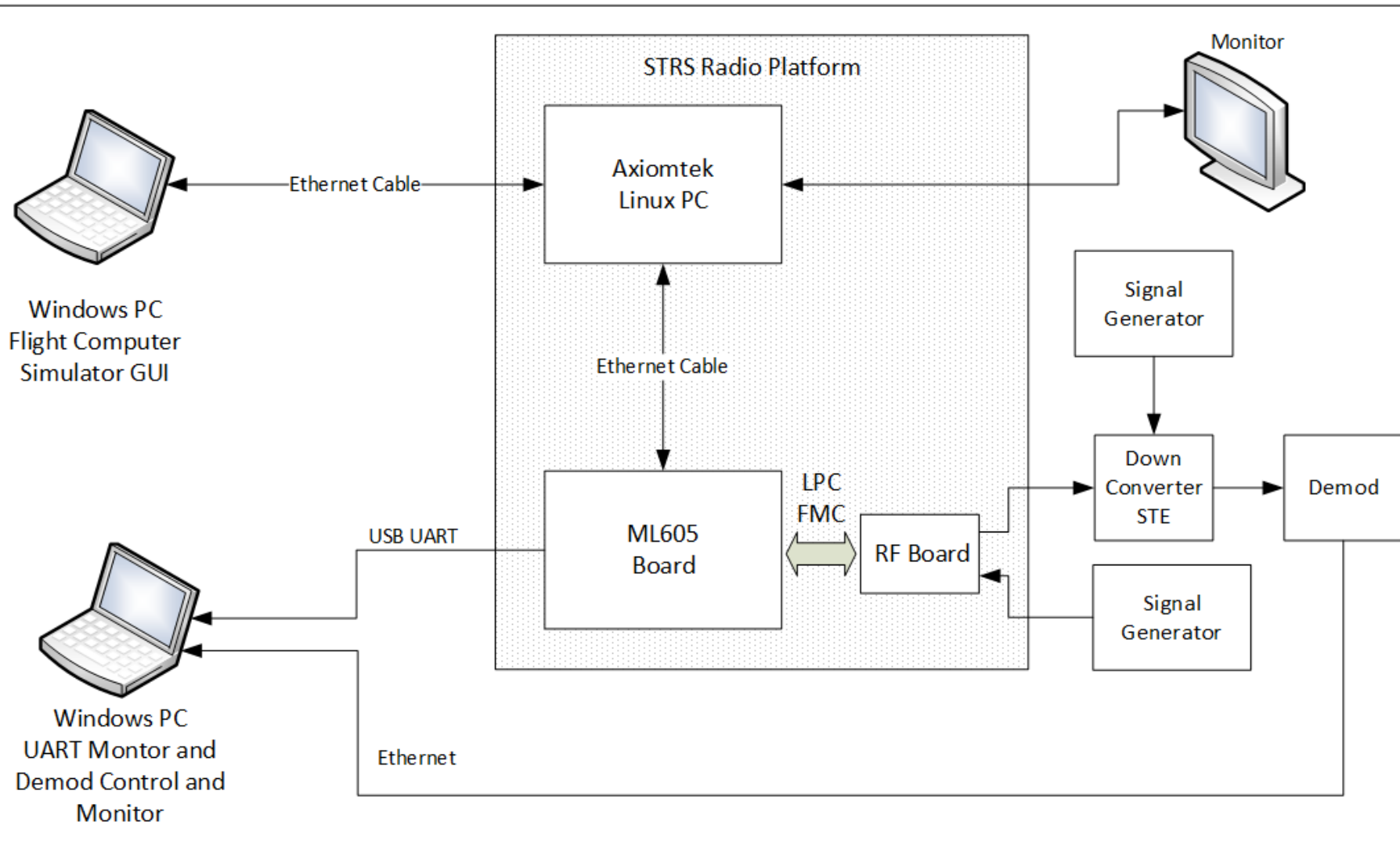
SDR Implementation

Testing

- The SDR was tested to verify that all the built-in functionality was working properly.
- A total of eight different tests were performed. Each of the tests demonstrated that each component interface was able to
 - transmit and receive data/commands without dropping packets.
- These tests verified that all the interfaces were working well together in a single direction and bi-directionally.
- The most important of these tests is the Full Data and Command Path SDR system, verification test with external sinks and sources.
 - Verified that all the interfaces of the SDR were working for commands and data simultaneously.
 - Verified that the SDR could receive data from outside sources and transmit that data to another outside destination without dropping packets.
 - Verified that the modulation scheme of the test waveform was working properly using encoded data from an outside source.

SDR Implementation

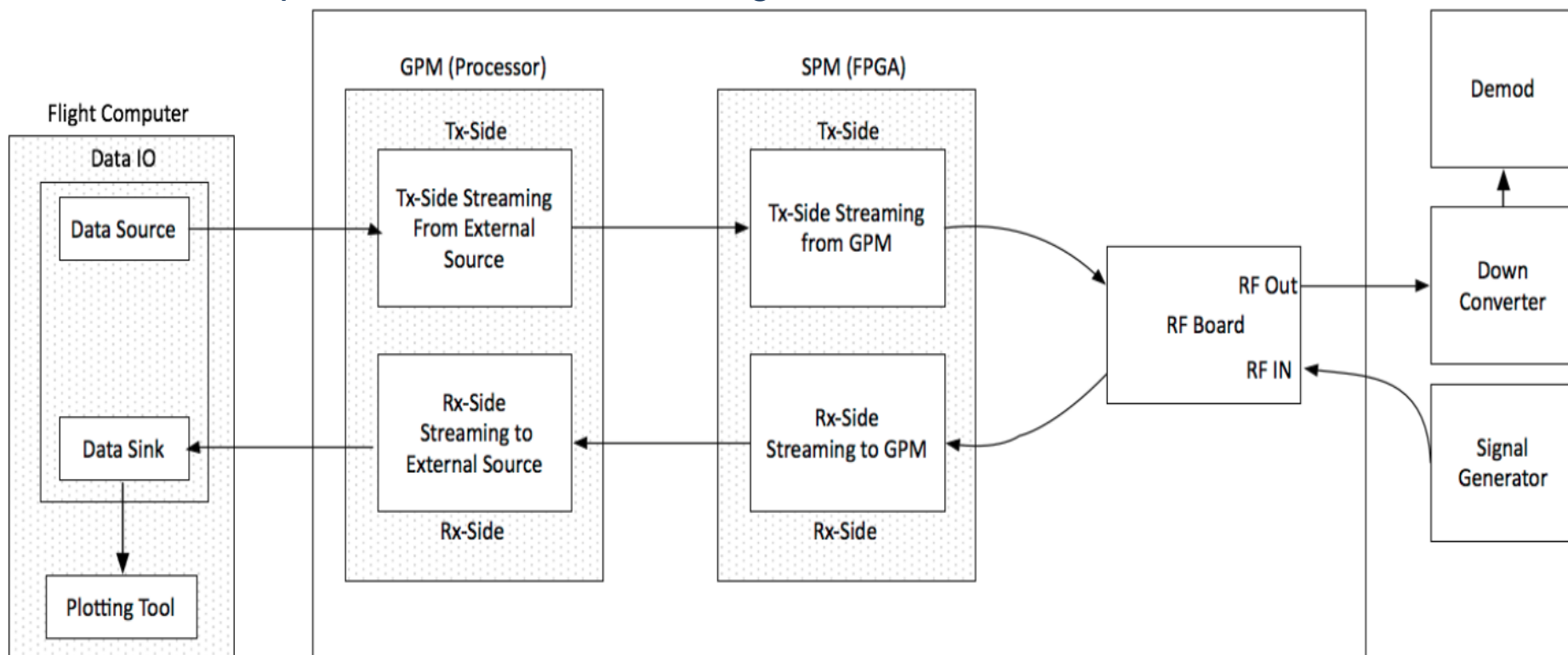
Testing: Space Radio Test Configuration



SDR Implementation

Testing: Data and Command Path Full Test

- The flight computer, signal generator, demodulator and down converter are peripherals independent of the SDR.
- The SDR is used as hardware in the loop to test the functionality of command and data paths in bidirectional configuration.





RESULTS & DISCUSSIONS

**Overview of Accomplished Work, Relevance to NASA's mission
and Overall Lessons Learned.**



Results

Overview of the Testing Outcomes

- The results from our verification testing show that the iPAS-STRS SDR is able to transmit, receive in simplex mode and in duplex mode without dropping packets or any evidence of bit errors.
- This indicates that all the control interfaces, software, firmware, RF and timing schemas between the modules (GPM to/from SPM to/from RF) are working properly.
- The successful capture of the transmitted sine wave shows that the system is able to decode the RF samples properly.
- Note that no demodulation is performed by the sample waveform since this is beyond the scope of a test waveform and is not necessary to demonstrate that the SDR is able to transmit and receive data from/to external sources as well as its internal sources.



Discussion

Inferences and Benefits of The Presented Work

- To conclude, we would like to highlight the benefits of using this platform as a development framework for STRS compatible, space communications waveforms.
- Standardization of software architectures across the SDR is a highly desirable feature in terms of software portability because by abstracting the hardware from the control interface, a developer can plug and play existing code. This speeds up development time.
- A developer writing waveforms on the iPAS-STRS SDR platform can simply bring their code as a new waveform, plug their low level control functions into the provided template and compile it under the STRS project. There is no need to be an expert on STRS to simply use it.
- The iPAS-STRS SDR is a step forward in the effort to infuse STRS into flight missions by providing a relatively simple framework for waveform development while demonstrating all the benefits of STRS on a practical application. This work serves as a development cradle for future space communication systems in next generation avionics and space flight SDRs.

